

A2

An embodiment of the invention is able to provide reliable resource management through the use of a joining function which enables the resource manager simply to join to the thread which is the current resource owner. Such a joining function is a standard language component of, for example, the Java<sup>TM</sup> language. Accordingly, a resource manager in accordance with an embodiment of the invention, can be provided with a single method "acquireDevice()" which may be called by any process, or application, wishing to use a resource (e.g., a telephony device such as a modem). Standard language primitives ensure that only one application may execute this method at one time, thereby guaranteeing exclusive access. The would-be owning application can identify its operating thread to the resource manager. By affecting the join on the thread which is the current resource owner, the resource manager is able simply to wait until the owning application thread terminates, at which time the join language function notifies the resource manager that the previously owning application has terminated. This notification is provided by the language constructs irrespective of the manner in which the previous owning application terminated. The resource manager is then able to allocate the resource to the requesting application.

**IN THE CLAIMS:**

Please amend claims 1, 3, 7, 9-12, and 16-18, as shown in the attached Appendix, to read as follows:

FINNEGAN  
HENDERSON  
FARABOW  
GARRETT &  
DUNNER LLP

1300 I Street, NW  
Washington, DC 20005  
202.408.4000  
Fax 202.408.4400  
[www.finnegan.com](http://www.finnegan.com)

1. (Amended) A resource manager operable to control allocation of a resource to competing computing processes including at least a first process and a second process,

Cott  
A3

the resource manager being responsive to identification of a thread for the first process requesting allocation of the resource, when the resource is already allocated to a thread for the second process, to establish a joining function to the thread for the second process, the joining function being operable to notify the resource manager on termination of the thread for the second process, and the resource manager being operable in response to termination of the thread for the second process to allocate the resource to the thread for the first process.

A4

3. (Amended) The resource manager of claim 2, wherein the first and second processes are software applications operable in the object oriented environment.

A5

7. (Amended) The resource manager of claim 1, wherein the join function is a join of the type provided in a Java language environment, and a language event passively releases a resource on termination of a thread identified by the join function.

A6

9. (Amended) The resource manager of claim 8, comprising a dispatch mechanism for controlling dispatching of a call received by the telephony device to the telecommunications applications.

FINNEGAN  
HENDERSON  
FARABOW  
GARRETT &  
DUNNER LLP

1300 I Street, NW  
Washington, DC 20005  
202.408.4000  
Fax 202.408.4400  
[www.finnegan.com](http://www.finnegan.com)

10. (Amended) A resource manager operable to control allocation of a resource to competing computing processes including at least a first process and a second process, the resource manager comprising means responsive to identification of a thread for the first process requesting allocation of the resource, when the resource is already allocated

to a thread for the second process, to establish a joining function to the thread for the second process and means responsive to the joining function notifying the resource manager on termination of the thread for the second process to allocate the resource to the thread for the first process.

*Cont  
A6*

11. (Amended) A computer software resource manager on a data carrier, the resource manager being operable to control allocation of a resource to competing computing processes including at least a first process and a second process, the resource manager being responsive to identification of a thread for the first process requesting allocation of the resource, when the resource is already allocated to a thread for the second process, to establish a joining function to the thread for the second process, the joining function being operable to notify the resource manager on termination of the thread for the second process, and the resource manager being operable in response to termination of the thread for the second process to allocate the resource to the thread for the first process.

12. (Amended) Telecommunications apparatus comprising at least one telephony resource for connection to a telecommunications network and a resource manager for controlling allocation of the telephony resource to competing computing processes including at least a first process and a second process, the resource manager being responsive to identification of a thread for the first process requesting allocation of the resource, when the resource is already allocated to a thread for the second process, to establish a joining function to the thread for the second process, the joining function being

FINNEGAN  
HENDERSON  
FARABOW  
GARRETT &  
DUNNER LLP

1300 I Street, NW  
Washington, DC 20005  
202.408.4000  
Fax 202.408.4400  
[www.finnegan.com](http://www.finnegan.com)

Cont  
p6

operable to notify the resource manager on termination of the thread for the second process, and the resource manager being operable in response to termination of the thread for the second process to allocate the resource to the thread for the first process.

16. (Amended) The telecommunications apparatus of claim 12, wherein the call processing applications comprise at least one application selected from:

- a call answering application;
- a voicemail application;
- a facsimile application; and
- a data application.

17. (Amended) A computer-implemented method of managing allocation of a resource to competing processes including at least a first process and a second process, the method including:

responding to identification of a thread for the first process requesting allocation of the resource, when the resource is already allocated to a thread for the second process, to establish a joining function to the thread for the second process;

responding to the joining function notifying termination of the thread for the second process to allocate the resource to the thread for the first process.

18. (Amended) The method of claim 17, wherein the join function is a join function of the type provided in a Java language environment, and a language event passively releases a resource on termination of a thread identified by the join function.